Received 8 November 2023; revised 11 January 2024; accepted 30 January 2024; date of publication 2 February 2024; date of current version 1 March 2024.

Digital Object Identifier 10.1109/TQE.2024.3361810

Rateless Protograph LDPC Codes for Quantum Key Distribution

ALBERTO TARABLE[1], RUDI PAOLO PAGANELLI[2], AND MARCO FERRARI[3] (Member, IEEE)

[1]Consiglio Nazionale delle Ricerche, Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, 10129 Torino, Italy
[2]Consiglio Nazionale delle Ricerche, Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, 40136 Bologna, Italy
[3]Consiglio Nazionale delle Ricerche, Istituto di Elettronica e di Ingegneria dell'Informazione e delle Telecomunicazioni, 20133 Milano, Italy

Corresponding author: Rudi Paolo Paganelli (e-mail: rudipaolo.paganelli@cnr.it).

This work was supported by European Union PON Ricerca e Innovazione 2014-2020 FESR /FSC under Project ARS01_00734 QUANCOM.

ABSTRACT Information reconciliation (IR) is a key step in quantum key distribution (QKD). In recent years, blind reconciliation based on low-density parity-check (LDPC) codes has replaced Cascade as a standard de facto since it guarantees efficient IR without a priori quantum bit error rate estimation and with limited interactivity between the parties, which is essential in high key-rate and long-distance QKD links. In this article, a novel blind reconciliation scheme based on rateless protograph LDPC codes is proposed. The rate adaptivity, essential for blind reconciliation, is obtained by progressively splitting LDPC check nodes, which ensures a number of degrees of freedom larger than puncturing in code design. The protograph nature of the LDPC codes allows us to use the same designed codes with a large variety of sifted-key lengths, enabling block length flexibility, which is important in largely varying key-rate link conditions. The code design is based on a new protograph discretized density evolution tool.

INDEX TERMS BB84, low-density parity-check (LDPC) codes, quantum key distribution (QKD), rateless codes.

I. INTRODUCTION AND RELATED WORK

Originating from the seminal work by Bennet and Brassard [1], which introduced the BB84 protocol, quantum key distribution (QKD) has become one of the most important applications of quantum physics to date. QKD exploits quantum laws to achieve unconditionally secure secret-key exchange at a distance. Thanks to this, no future advance in technology will be able to threaten the security guarantees obtained by QKD. In recent years, QKD has leaped from the laboratory to real-world security applications.

In a general QKD setting, Alice and Bob want to share a secure shared key for cryptographic applications, such as encryption and authentication, on an insecure classical channel. In discrete-variable QKD (DV-QKD), which is the most common implementation, Alice sends a sequence of qubits carried by photons to Bob through a quantum channel, which is measured by Bob. Through public discussion on a classical channel, Alice and Bob select a subsequence of qubits, each one obtaining a classical binary random sequence called a *sifted key*.

Based on quantum laws, in ideal conditions, the sifted key would be identical on the two sides. In practice, Bob's sifted key may differ from Alice's one because of various factors, such as nonidealities of devices or intervention of an eavesdropper (Eve), which attempts to learn the qubit values, thus causing disturbance to the traveling photons. Because of that, there is a nonzero probability, called the quantum bit error rate (QBER), that a key bit is flipped in Bob's sifted key. After the key-distribution phase, a classical phase follows, whose twofold goal is to correct all discrepancies between the sifted keys at the two sides *information reconciliation* and to null out all information that Eve may have gleaned (*privacy amplification*).

Information reconciliation (IR) is the subject of this article. In its various implementations, the Cascade protocol, introduced in [2], is the most popular IR protocol, with a typically good efficiency, especially for low QBER. The algorithm is organized into successive phases. In each phase, Alice reveals the parities of blocks of key bits. Whenever Bob observes a discrepancy in the parity of one block, a dichotomic search is performed in order to correct one error. Every error observed in a later phase implies a cascade of dichotomic searches in previous phases, where it had been masked by other errors, hence the name. The disadvantage of Cascade lies in the typically large number of required communication rounds, leading to a relevant communication overhead, in particular when the algorithm parameters are optimized for the current QBER to achieve the largest possible efficiency [3].

Later, forward-error-correction (FEC) schemes have been proposed for application to IR. Such methods rely on FEC such as Hamming [4], low-density parity-check (LDPC) [5], or Polar codes [6] to avoid the high interactivity of Cascade. Particularly, in [7], a blind LDPC-based protocol allows us to achieve IR without knowing the channel QBER at the beginning. The number of communication rounds is dramatically reduced with respect to Cascade. The price to pay is the typically slightly lower efficiency of the scheme, whose loss depends on the length of the sifted key. The protocol in [7] has then been improved in [8], by symmetrizing the operations performed by Alice and Bob. Subsequently, Borisov et al. [9] have improved the scheme in [7], by refining the starting QBER estimate and optimizing the amount of information disclosed at each round, yielding almost the same performance as in [8].

In [10], standard "off-the-shelf" LDPC codes are employed for IR with the additional use of data reordering and bit filling to combat error bursts and to increase the error correction capability of the codes for high-QBER QKD links. However, the algorithm is highly interactive and the average inefficiency is not shown. Several recent papers, for instance [11] and [12], have proposed variants of IR-LDPC without real advantages in terms of reduced inefficiency or interactivity compared to the work in [8]. It is worth noting that while the scheme in [12] is claimed to improve on the symmetric scheme in [8], we believe that the inefficiency computation in [12] is somewhat imprecise, as it will be specified later, thus making the comparisons unreliable.

The important feature of IR-LDPC is in the reduced interactivity w.r.t Cascade, which is crucial in future high key-rate QKD links. The main credit of the work in [7] and [8] is to partly join this feature with a good coding efficiency thanks to rate-adaptivity and symmetric reconciliation. However, there may be scenarios where there is no symmetry between sender and receiver and an intrinsically symmetric scheme may transform the party with limited computational power into a bottleneck for the QKD link. A typical example is satellite QKD in which the satellite station has limited computational resources onboard [13].

Besides, the works in [7] and [8], as all the efficient proposals of IR-LDPC presented so far, are based on the irregular LDPC ensembles described in [5]. These ILDPC codes have very good convergence thresholds at the price of large node degrees and they are hard to design, in particular with short codes. In addition, if different key lengths are needed to adapt to application requests or QKD link conditions, codes of different block lengths drawn from the same ensemble must be designed independently, ending with a large set of codes designed for each code rate and block length.

In this article, we propose a new asymmetric IR scheme based on rateless protograph LDPC codes. Differently from the work in [7], where rate adaptivity is based on the revelation by Alice of a fraction of punctured bits, in our scheme, the coding rate is adapted online to the channel by adding parity-check equations in every communication round, similarly to Cascade. Because of that, our scheme bears some similarity to Fountain codes, also known as rateless codes. With respect to the schemes in [7] and [8], rateless protograph LDPC codes show a better efficiency for a wide range of QBER values and an easier and more flexible design able to adapt to different sifted-key block lengths. Finally, they require a lower number of communication rounds to achieve the key reconciliation. The rateless IR scheme is based on families of protograph LDPC codes that are optimized via protograph DDE, which is an innovative design tool and an original contribution of our article.

The rest of the article is organized as follows. In Section II, we describe the reference QKD scenario. In Section III, we summarize the different LDPC-based IR protocols taken from the literature. In Section IV, we introduce our proposal, rateless LDPC codes. In Section V, we show simulation results, highlighting the advantages of our proposal. Finally, Section VI concludes this article.

NOTATION

In this article, lowercase boldface is used for (column) vectors. The ith entry of vector $\mathbf{v}$ is denoted v_i. Given a binary operator $\odot$, in general nonassociative, we adopt the notation $\bigodot_{i=1}^{n} a_i = a_1 \odot (a_2 \odot \ldots (a_{n-1} \odot a_n))$. Moreover, we use $a^{\odot n}$ to mean $\bigodot_{i=1}^{n} a$.

II. SYSTEM DESCRIPTION

Alice and Bob need to create a secure key, which can then be used for encryption. To do this via QKD, they share an authenticated, insecure, noiseless classical channel (e.g., a conventional TCP/IP link) and a quantum channel. A practical implementation of this quantum channel could be an optical fiber, where qubits are encoded in the polarization of photons.

We assume that Alice and Bob implement the BB84 scheme with polarization encoding, although our IR algorithm can be applied to any QKD protocol generating correlated binary strings as, for instance, decoy-state BB84 protocols [14] or entanglement-based QKD [18]. Let $\mathcal{B} = \{|0\rangle, |1\rangle\}$ be a reference orthonormal basis for encoding a bit value (0 or 1, respectively) with the polarization of a single photon (a qubit). Let $\mathcal{B}' = \{|+\rangle, |-\rangle\}$ be another basis, obtained by Hadamard transformation, i.e.,

$$|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}. \quad (1)$$

In the Hadamard basis, the bit value 0 is associated with $|+\rangle$ and the bit value 1 is associated with $|-\rangle$.

In BB84, Alice performs the following three steps several times.

1) She picks at random one of the two bases.
2) Given the basis, she picks at random one logical bit and she selects the associated basis vector.
3) She polarizes a single photon in the selected basis vector and sends it to Bob through the quantum channel.

For each received photon, Bob picks at random one of the two bases and measures its polarization state. After the distribution phase is over, Alice reveals the sequence of bases for the qubits received by Bob. Then, they throw away all qubits for which the basis measured by Bob is not the same as the one chosen by Alice, thus remaining with a shortened qubit sequence, which is mapped into a classical bit sequence, called the *sifted key*.

Ideally, Alice's and Bob's sifted keys are identical sequences of classical bits but there may be mismatches between their keys because of nonidealities in the projection process and in the detection. Thus, we model the errors in the quantum part of the protocol as a binary symmetric channel (BSC), that is, Bob's sifted key is identified with a length-N binary vector[1]

$$\mathbf{y} = \mathbf{x} + \mathbf{e} \tag{2}$$

where the sum is modulo-2, $\mathbf{x}$ is Alice's sifted key and $\mathbf{e}$ is a length-N error vector, composed of i.i.d. binary random variables. In case of asymmetric settings, the symmetry of the BSC can always be enforced, at the cost of one additional communication round, by adding a uniformly random bit string $\mathbf{r}$ to both $\mathbf{x}$ and $\mathbf{y}$ immediately after the sifting procedure. Let $\epsilon = \mathbb{P}\{e_i = 1\}$ be the QBER, assumed constant for the whole key block. In ideal, unperturbed conditions, $\epsilon = 0$. However, nonidealities of the system (misalignment of Bob's measurement device, physical-channel nonlinear effects, crosstalk, dark counts, etc.), as well as perturbations induced by Eve, cause $\epsilon > 0$. In this case, the channel capacity of the BSC is $C = 1 - h_2(\epsilon)$ where $h_2(\epsilon)$ is the binary entropy and measures in bits the amount of information lost for each channel use.

The purpose of IR is precisely to recover $\mathbf{x}$ on Bob's side. This typically happens through Alice's partial revelation of information about $\mathbf{x}$ through the insecure, classical channel. If IR succeeds, a successive phase of privacy amplification allows nulling out all the information that Eve may have acquired on $\mathbf{x}$. If IR does not succeed, the sifted key is thrown away, and a new distribution phase can start.

III. LDPC-BASED IR

A. LDPC CODES

LDPC codes are linear block codes that can reach a performance very close to the theoretical Shannon limit on general binary input channels while entailing a practically affordable complexity of encoding and decoding [15]. A binary LDPC code $\mathcal{C}$ of length N_v with N_c check nodes (CNs) and dimension K is characterized by an $N_c \times N_v$ sparse binary parity-check matrix $\mathbf{H}$. Code $\mathcal{C}$ is defined as the null space of the columns of $\mathbf{H}$, i.e.,

$$\mathbf{c} \in \mathcal{C} \Longleftrightarrow \mathbf{H}\mathbf{c} = \mathbf{0}. \tag{3}$$

If the matrix $\mathbf{H}$ is full-rank, i.e., its rows are all linearly independent, the number of *information bits* carried by each codeword is $K = N_v - N_c$ since it is the number of (binary) degrees of freedom in the choice of the codeword to be transmitted.

The ratio $R = K/N$ between the number of information bits K and the number N of bits transmitted through the channel (the number of channel uses), is called *code rate* and it is bounded earlier by the channel capacity for error-free transmission [15]. If all the LDPC codeword bits are transmitted, $N = N_v$.

When a subset of the transmitted bits match the information bits, the code is called *systematic*. A systematic codeword is, thus, the concatenation of K information bits with $N - K$ additional parity bits, which are sent to the decoder to recover the errors that occurred during transmission.

When the channel capacity is much larger than K/N_v, an easy way of increasing the code rate is to puncture some of the bits, say N_p, i.e., to avoid transmitting them. In this way, the number of channel uses is reduced to $N = N_v - N_p$ and the code rate is increased to $R = K/(N_v - N_p)$. From another point of view, the number of parity bits transmitted is reduced when the channel quality is larger and the whole set of $N_v - K$ is not needed.

An alternative way to characterize the LDPC code is through its Tanner graph, which is a bipartite graph with N_v variable nodes (VNs) $\{v_1, \ldots, v_{N_v}\}$, corresponding to the columns of $\mathbf{H}$, and N_c CNs $\{c_1, \ldots, c_{N_c}\}$, corresponding to the rows of $\mathbf{H}$. The Tanner graph of $\mathcal{C}$ is the adjacency graph of $\mathbf{H}$, i.e., there is an edge between c_i and v_j if and only if the (i, j) entry of $\mathbf{H}$ is 1. In such a case, c_i and v_j are said to be neighbors of each other.

Normally, the LDPC decoder is based on belief propagation (BP), in which VNs and CNs exchange messages along the edges of the Tanner graph for a certain number of iterations. At each iteration, all VNs in the Tanner graph send messages to their neighbor CNs. In their turn, CNs reply with their own messages to their neighbor VNs. The BP decoder with initialization and update of the messages exchanged and the decisions made based on these messages are described in detail in Appendix A.

B. RATE-ADAPTIVE LDPC-BASED IR

In this section, we give a unified description of LDPC-based IR that can encompass the schemes in [7] and [8] and our proposal. First of all, we remark that the length of the sifted key N matches the number of qubits measured by Bob in Alice's preparation basis, whereas the LDPC block length is

[1] In case the sifted key is excessively long, or it is a stream of data, it can be divided into subblocks. In this case, N is the length of each of these blocks.

$N_v = N + N_p$ in case of *punctured* bits, i.e., bits that have not been encoded in qubits, although they are part of the codeword.

Let $\mathbf{x}_e$ be the extended sifted key on Alice's side of length $N_v = N + N_p$, including the sifted key $\mathbf{x}$ and the punctured bits, which can be picked randomly by Alice, for instance using a quantum random number generator, in the same way as she does for the basis or the photon polarization of each qubit. In LDPC-based IR, after the distribution phase, Alice sends to Bob $N_{c,0}$ syndrome bits, i.e., the results of the $N_{c,0} \times N_v$ parity-check matrix $\mathbf{H}_0$ of a "mother" LDPC code, applied to the N_v random bits composing $\mathbf{x}_e$. The knowledge of the length-$N_{c,0}$ syndrome $\mathbf{s}_0 = \mathbf{H}_0\mathbf{x}_e$ transforms the unconstrained sequence $\mathbf{x}_e$ into a codeword that belongs to a coset code $\mathcal{C}_0$ for which

$$\mathbf{c} \in \mathcal{C}_0 \iff \mathbf{H}_0\mathbf{c} = \mathbf{s}_0. \tag{4}$$

The matrix $\mathbf{H}_0$ is known to Bob, who, on the basis of the channel output, tries to decode $\mathbf{x}_e$ using the BP algorithm. If decoding converges, he sends an acknowledge (ACK) message to Alice on the noiseless classical channel. In this way, Alice and Bob will share, with high probability, the same extended sifted key $\mathbf{x}_e$. Otherwise, Bob sends a not-ACK (NACK) message to Alice on the classical channel. She will then send an additional set of Δ syndrome bits that will reduce the code rate and can prove useful on Bob's side to obtain decoder convergence.

Thus, in general, LDPC-based IR consists of a number of successive rounds, where at round $n = 0, 1, 2 \ldots$, the correct extended sifted key $\mathbf{x}_e$ belongs to the coset code $\mathcal{C}_n$ defined by

$$\mathbf{c} \in \mathcal{C}_n \iff \mathbf{H}_n\mathbf{c} = \mathbf{s}_n \tag{5}$$

where for $n > 0$

$$\mathbf{H}_n = \left[\mathbf{H}_{n-1}^{\mathsf{T}}, \widetilde{\mathbf{H}}_n^{\mathsf{T}}\right]^{\mathsf{T}} \tag{6}$$

$$\mathbf{s}_n = \left[\mathbf{s}_{n-1}^{\mathsf{T}}, \widetilde{\mathbf{s}}_n^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{7}$$

$\widetilde{\mathbf{H}}_n$ and $\widetilde{\mathbf{s}} = \widetilde{\mathbf{H}}_n\mathbf{x}_e$ being the $\Delta \times N_v$ matrix of the additional checks on the rows and the new length-Δ syndrome portion sent by Alice at round n, respectively. Notice that matrix $\mathbf{H}_n$ has $N_{c,0} + n\Delta$ rows and N_v columns. Thus, the code rate at round n, $n \geq 0$, is given by

$$R_n = \frac{N_v - N_{c,0} - n\Delta}{N}. \tag{8}$$

The rate R_0 of the mother code can be chosen according to an estimate of the QBER ϵ.

In Fig. 1, a block diagram summarizes the flow of the operations described for rate-adaptive reconciliation.

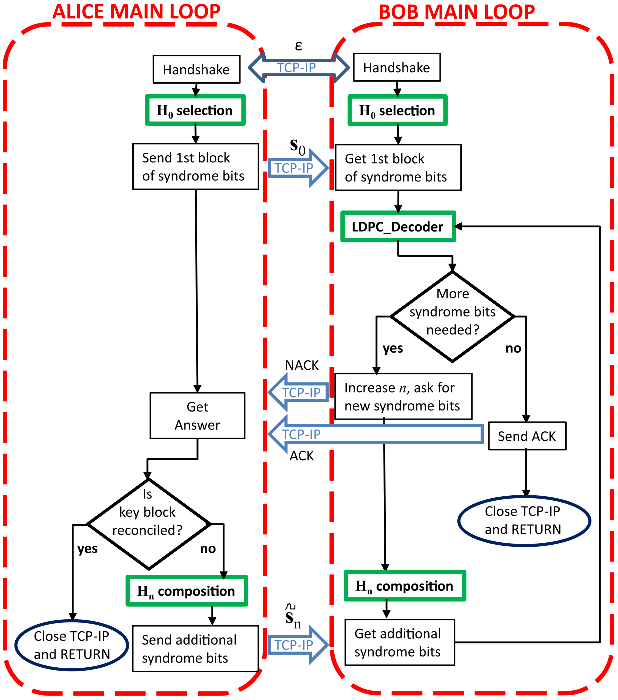

FIGURE 1. **Block diagram of the rate-adaptive reconciliation procedure run by Alice and Bob.**

It is customary to measure the performance of the IR schemes through the *coding inefficiency*, which is defined by the ratio between the number of information bits about the secret key revealed on the public channel (i.e., the number of transmitted syndrome bits that were necessary for decoder convergence) and the average information loss in the process generating the sifted keys. If decoding succeeds at round n, the inefficiency is given by

$$f_n = \frac{N_{c,0} - N_p + n\Delta}{N h_2(\epsilon)} = \frac{1 - R_n}{h_2(\epsilon)}. \tag{9}$$

Note that the number N_p of punctured bits is subtracted from the number of information bits revealed because they will be recovered as part of the reconciled key. The punctured bits are at least as secure as the sifted-key bits since Eve cannot obtain any information about them during the quantum information exchange. The average realized inefficiency [where the average is taken with respect to $\mathbf{e}$ in (2)] will then be given by

$$\overline{f} = \frac{1 - \overline{R}}{h_2(\epsilon)} \tag{10}$$

where $\overline{R}$ is the average rate. The choice of Δ is a tradeoff between two opposite demands: Decreasing Δ will typically increase the average efficiency but at the price of a larger average number of communication rounds.

C. BLIND RECONCILIATION

The blind scheme in [7] fits in the description of LDPC-based IR given in the previous section, by adequately defining the additional syndrome bits sent by Alice at Bob's request. The case $\Delta = 1$ is considered in [7]. After the first syndrome transmission, at each subsequent round, if needed, Alice reveals the value of a punctured bit. It is tantamount to say that

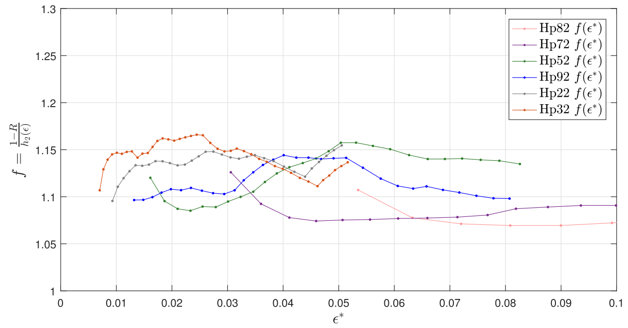

FIGURE 2. **Inefficiency *f* of each member of each designed rateless-LDPC family, computed according to each code rate *R* and convergence threshold ϵ^*.**

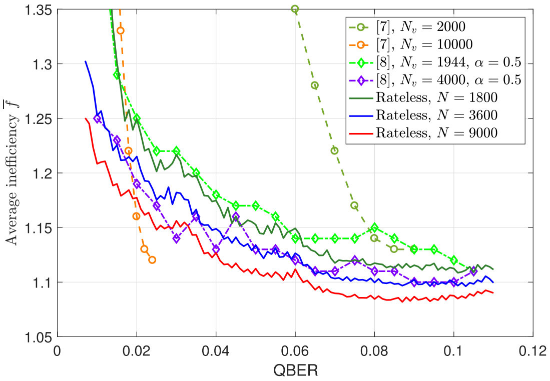

FIGURE 3. **Average inefficiency $\bar{f}$ with the proposed rateless LDPC scheme, for certain sifted-key lengths, in comparison with the schemes in [7] and [8] with similar block lengths.**

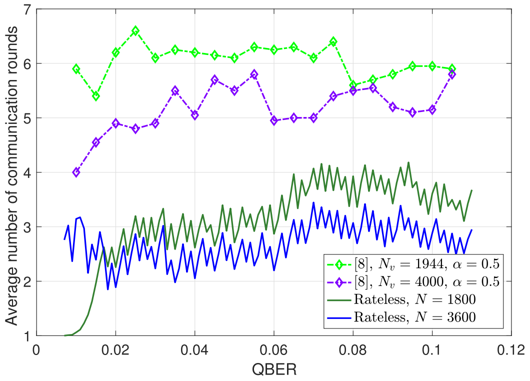

FIGURE 4. **Average number of communication rounds needed to reconcile the key with the proposed rateless LDPC scheme, for certain sifted-key lengths, in comparison with the scheme in [8] with similar block lengths.**

the additional syndrome bit is computed as the parity of that punctured bit.

A shortcoming of this construction is that the rounds are over when all punctured bits are revealed. However, the advantage with respect to a static, rate-adaptive scheme is apparent from [7, Figs. 5–7].

D. SYMMETRIC BLIND RECONCILIATION

In [8], a symmetric blind reconciliation procedure is proposed as an improvement of the scheme in [7]. The symmetric scheme takes its name from a different schedule of the messages sent by Alice and Bob. By exchanging both syndrome vectors, Bob and Alice perform decoding on equivalent input, so that they observe the same results. This has a beneficial effect on the communication rounds since Alice does not have to wait for Bob's request to send further syndrome bits. From the point of view of efficiency, the improvement of the symmetric scheme arises from the choice of the bits to be revealed at each communication round. They are not necessarily punctured bits, and they are chosen as the ΔVNs that have obtained the smallest output LLR magnitudes among all in the previous decoding stage. Results reported in [8] show that the symmetric scheme improves the efficiency with respect to the blind scheme up to an average of 10% and the number of communication rounds up to 30%.

In [12], a layered coding scheme is proposed, in which syndrome bits of the upper layers are encoded into punctured bits of lower layers. Mao et al. [12] claim that their scheme improves the IR inefficiency with respect to the work in [8]. However, their evaluations do not take into account that, with such a layered coding scheme, the reconciled key bits are not independent anymore; hence, the inefficiency curves shown in [12] are somewhat imprecise.

IV. RATELESS LDPC IR

In this section, we describe our proposed rateless protograph LDPC IR scheme. Unlike the symmetric scheme in [8], the sequence of parity-check matrices $\{\mathbf{H}_n\}_n$ are defined a priori, to avoid a double exchange of information between Bob and Alice, and a doubled computation overload for decoding. In the next section, we first introduce protograph LDPC codes, upon which our rateless LDPC codes are based.

A. PROTOGRAPH LDPC CODES

Protograph LDPC codes are LDPC codes whose Tanner graph is obtained by lifting (expanding) a smaller bipartite graph, called protograph, characterized by ν_c CNs $\{\kappa_1, \ldots, \kappa_{\nu_c}\}$ and ν_v VNs $\{\omega_1, \ldots, \omega_{\nu_v}\}$, out of which ν_p are punctured. The protograph is characterized by a $\nu_c \times \nu_v$ adjacency matrix $\mathbf{T}$, with nonnegative integer entries, whose (i, j) entry t_{ij} represents the number of edges connecting κ_i to ω_j (parallel edges are permitted). To obtain the Tanner graph of the LDPC code, we first fix the lift factor S, which is a large integer. Then, the following steps are performed.

1) Replicate the protograph S times, obtaining a (disconnected) graph with $N_c = \nu_c S$ CNs, $N_v = \nu_v S$ VNs, out of which $N_p = \nu_p S$ are punctured; the S CNs obtained as replicas of κ_i will be denoted $(\kappa_i, 1), \ldots, (\kappa_i, S)$ and analogously for the VNs; there will be t_{ij} edges connecting (κ_i, s) and (ω_j, s) for $s = 1, \ldots S$.
2) Randomly permute the tips of the edges in the graph; precisely, for every i and j, if $t_{ij} > 0$, pick

t_{ij} random permutations $\pi_{ij}^{(1)}, \ldots, \pi_{ij}^{(t_{ij})}$ of the integers $\{1, \ldots, S\}$; then, connect (κ_i, s) to the t_{ij} VNs $\{(\omega_j, \pi_{ij}^{(1)}(s)), \ldots, (\omega_j, \pi_{ij}^{(t_{ij})}(s))\}$.

In practice, the random permutations are constrained to be circular shifts of the identity permutation, so that each of them can be derived from the value of the corresponding shift, allowing a substantial saving in memory storage. For example, the shift σ corresponds to the permutation $\pi(s) = \text{mod }(s + \sigma - 1, N) + 1, s = 1, \ldots, S$.

There are several advantages of protograph-based LDPC codes. First, protograph LDPC codes lend themselves easily to a parallel implementation of the decoder, where the degree of parallelism can be chosen as large as the lift factor.

The second, most important advantage for IR applications, is that the same protograph can be used to obtain LDPC codes with different block lengths by changing S; thus, allowing a considerable flexibility in the dimension of the sifted key without the need of a new code design for each length.

Finally, an important advantage comes in the phase of code design. Capacity-achieving LDPC codes are the *irregular* ones (ILDPC), i.e., those characterized by VNs and CNs with different degrees (number of neighbors) in the Tanner graph. Roughly speaking, large-degree VNs converge faster and then help lower-degree VNs in convergence. Large-degree CNs keep the code rate high but low-degree CNs propagate stronger messages that accelerate convergence. However, code irregularity must be carefully studied to improve the decoder convergence.

A common approach is to design a *code ensemble* with good *average* convergence properties. Two irregularity profile polynomials $\lambda(x)$, $\rho(x)$ and a code length N_v identify a code ensemble, which is the set of all Tanner graphs with N_v VNs and edge distribution[2] that matches $\lambda(x)$, $\rho(x)$ [15]. The ensemble convergence threshold for the BP algorithm can be computed via density evolution (DE), under the approximation of independence between messages exchanged during the iterations, averaging on the Tanner graphs belonging to the ensemble. As N_v gets large, all the Tanner graphs in the ensemble tend to perform in the same way, and any graph can be picked at random from the ensemble (or by pseudorandom choice) obtaining a well-designed code. This approach has been followed for IR in [5] to design ILDPC ensembles analyzed through discretized DE (DDE) [16] and those polynomials have then been used in [7] and [8] to design the ILDPC codes.

The previously described designed method, however, does not work so well when the block length is small, in particular when large node degrees are used to improve the convergence threshold. Besides, it penalizes ensembles with punctured and/or degree-1 VNs in the Tanner graph since their presence may heavily affect the ensemble average decoding convergence. Nevertheless, when properly connected and inserted in the graph, these nodes can be exploited to design codes with excellent convergence thresholds [17]. Such a control can be achieved by careful protograph selection since the properties of the BP decoder for the LDPC code can be studied directly by performing BP decoding on the protograph from which it is expanded. Although, in principle, the analysis could be carried out directly on the expanded graph, it is feasible only when the number of edges in the graph is small enough since DDE needs to track the evolution of two message probability mass functions (pmfs) for each edge.

In this article, we design good protograph codes for IR, by extending the DDE method [16] to quantized BP decoding on a protograph, which is described in Appendix B. For $S \to \infty$, the LDPC decoder performance is characterized by a threshold ϵ^*, below which the LDPC decoder always converges, and above which it never converges. The value of ϵ^* can be computed by protograph DDE, which is described in the next section.

B. PROTOGRAPH DDE

DDE is a technique conceived to analyze the convergence of BP decoders, in which the messages exchanged by CNs and VNs are quantized to a set of discrete levels, as it is in the practical implementations of the LDPC decoder. DDE tracks the pmfs of the BP messages along the iterations for $S \to \infty$. In this infinite block length limit, at each node, the incoming messages can be assumed to be independent random variables. DDE gives a good approximation of the BP decoder behavior also for large but finite block length. In the following, we give a description of DDE applied to a protograph, which is an original contribution of this article.

Suppose that $\mathbf{x}_e = \mathbf{0}$, i.e., $x_j = 0$ for all j, so that $\mathbf{y} = \mathbf{e}$. Since the BSC is a symmetric channel, this hypothesis does not entail any loss of generality. Consider BP decoding on a protograph, with quantization function $\mathcal{Q}$ on a set of levels $\{q_m\}_{m=-M}^{M}$, as described in Appendix B. Define $\mathcal{P}_{j,\text{ch}}[m] = \mathbb{P}\{\mathcal{Q}(L_{j,\text{ch}}) = q_m\}$ as the pmf of the channel LLR for VN ω_j, for $m = -M, \ldots, M$. This pmf is the input of the DDE algorithm and for the unpunctured VNs reads

$$\mathcal{P}_{j,\text{ch}}[m] = \begin{cases} 1-\epsilon, & q_m = \mathcal{Q}\left(\log \frac{1-\epsilon}{\epsilon}\right) \\ \epsilon, & q_m = -\mathcal{Q}\left(\log \frac{1-\epsilon}{\epsilon}\right) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

whereas for the punctured VNs $\mathcal{P}_{j,\text{ch}}[0] = 1$.

Let $\mathcal{P}_{i\leftarrow j}(\ell)$ (respectively, $\mathcal{P}_{i\to j}(\ell)$) denote the pmf of the message from ω_j to κ_i (respectively, from κ_i to ω_j) at iteration ℓ. Notice that these pmfs depend on the iteration since the messages are computed according to the node inputs, which vary with iterations. Besides, parallel edges have the same pmf. At the beginning, the CN messages are initialized to 0 for all i and j, thus $\mathcal{P}_{i\to j}(0)[0] = 1$. At iteration $\ell = 1, 2, \ldots$, the pmf of the message output by each node can be updated assuming independent input messages, according to

[2] The most popular way to assign irregularity profiles is by the polynomial coefficients λ_i and ρ_i as the fractions of edges connected to degree-i VNs and CNs, respectively.

TABLE 1. Rateless LDPC Code Families, Together With the Protograph Properties for the Mother Code $\mathcal{C}_0$

Name	R	ν_c	ν_v	ν_p	ϵ^*	$(1-R)/h_2(\epsilon^*)$
Hp32	14/15	15	155	5	0.0070	1.1068
Hp22	11/12	15	125	5	0.0093	1.0955
Hp92	8/9	15	95	5	0.0132	1.0965
Hp52	13/15	15	80	5	0.0161	1.1201
Hp72	7/9	15	50	5	0.0306	1.1260
Hp82	2/3	15	35	5	0.0535	1.1071

TABLE 2. Lift Factor (S) for the Different Rateless LDPC Code Families and for Different Sifted-Key Block Lengths (N)

N	1800	3600	9k	90k
Hp32	NA	24 (6)	60 (6)	600 (6)
Hp22	NA	30 (6)	75 (6)	750 (8)
Hp92	NA	40 (6)	100 (6)	1000 (8)
Hp52	24 (6)	48 (6)	120 (6)	1200 (8)
Hp72	40 (6)	80 (6)	200 (8)	2000 (8)
Hp82	60 (6)	120 (8)	300 (8)	3000 (8)

the following equations:

$$\mathcal{P}_{i\leftarrow j}(\ell) = \mathcal{V}_{ij}\left(\mathcal{P}_{j,\text{ch}}, \{\mathcal{P}_{i'\rightarrow j}(\ell-1)\}_{\kappa_{i'}\in\mathcal{N}_v(j)}\right) \tag{12}$$

$$\mathcal{P}_{i\rightarrow j}(\ell) = \mathcal{C}_{ij}\left(\{\mathcal{P}_{i\leftarrow j'}(\ell)\}_{\omega_{j'}\in\mathcal{N}_c(i)}\right). \tag{13}$$

Functions $\mathcal{V}_{ij}$ and $\mathcal{C}_{ij}$ are derived in Appendix C. The pmf of the output LLR at iteration ℓ is computed as

$$\mathcal{P}_j(\ell) = \mathcal{O}_j\left(\mathcal{P}_{j,\text{ch}}, \{\mathcal{P}_{i\rightarrow j}(\ell)\}_{\kappa_i\in\mathcal{N}_v(j)}\right) \tag{14}$$

where the expression of $\mathcal{O}_j$ is also given in Appendix C. We can compute the error probability at iteration ℓ for ω_j as the probability that the output LLR is negative,[3] i.e.

$$P_{e,j}(\ell) = \sum_{m<0}\mathcal{P}_j(\ell)[m] + \frac{1}{2}\mathcal{P}_j(\ell)[0]. \tag{15}$$

The threshold ϵ^* is the infimum of all ϵ for which

$$\lim_{\ell\to\infty} P_{e,j}(\ell) = 0 \qquad \forall j. \tag{16}$$

We remark once again that the previous analysis, which tracks at each iteration the pdfs $\mathcal{P}_{i\leftarrow j}$, $\mathcal{P}_{i\rightarrow j}$ $\quad\forall v_i, c_j$, is only feasible when the code graph is built as the expansion of a small protograph, whose number of CNs and VNs determines the analysis complexity, independently of the lift factor and, thus, of the block length.

C. DESIGN OF THE MOTHER CODE

In our rateless protograph LDPC IR scheme, the design of the mother code $\mathcal{C}_0$ is based on accumulate–repeat–jagged–accumulate (ARJA) codes, in particular on their AR6JA variants [17]. The AR6JA codes are a family of protograph LDPC codes with rates ranging from 1/2 to 9/10 and with excellent performance on the AWGN channel, where the loss with respect to capacity is typically within 0.1 dB. In order to cover the whole range of QBER values, we have picked some AR6JA codes and we have modified them to optimize their thresholds for the BSC.

Table 1 shows the parameters of the six protographs obtained, with their respective code rates, number of CNs ν_c and VNs ν_v, number of punctured VNs ν_p, BSC threshold ϵ^* and ratio between code rate R and channel capacity at threshold $1-h_2(\epsilon^*)$. Each protograph will correspond to the mother code $\mathcal{C}_0$ for a certain QBER range. The matrix $\mathbf{H}_0$ will then be obtained by lifting the protograph with a lift factor S. We have designed lift factors and shifts for four distinct lengths N of the sifted key. Such values of N were chosen to be compatible with all protograph sizes, and this requires N to be a multiple of 1800. Thus, $N = 1800$ is the minimum possible key length. Then, by increasing N, it is possible to obtain a larger graph girth by properly designing the shifts, as shown in Table 2. Table 2 shows the four chosen key lengths and the corresponding values of S for the different families that have been designed. For each protograph and lift factor, we have designed the shifts by maximizing, via a pseudorandom algorithm, the code girth G, defined as the minimum cycle size in the Tanner graph of the code. Table 2 also shows the achieved girth with all combinations of code family and key length. Note that the shifts designed for the lift factor S can be used also for lift factors $S' > S$, thus extending the possible values of the sifted-key length at no computational cost.

As can be seen, for certain families a key length $N < 3600$ is not possible, as it would require a too small value of S.

D. CHECK SPLITTING FOR RATE ADAPTATION

Matrix $\mathbf{H}_n$ is obtained from matrix $\mathbf{H}_{n-1}$ by *check splitting*. In the following, we describe check splitting on the protograph. Let $\mathbf{T}_{n-1}$ be the adjacency matrix of the protograph corresponding to $\mathbf{H}_{n-1}$ and let $\mathbf{t}$ be one of its rows. Matrix $\mathbf{T}_n$ is then obtained by substituting $\mathbf{t}$ with the two rows $\mathbf{t}'$, $\mathbf{t}''$, where[4] $\mathbf{t}' + \mathbf{t}'' = \mathbf{t}$. Splitting one row in the protograph corresponds to splitting S rows in the parity-check matrix.

In order to generate the children codes $\{\mathcal{C}_n\}$, $n > 0$, of a given family, we have resorted to a greedy algorithm of choosing the check to split, and the way of splitting it. A straightforward way of generating the family members could be to choose the check splitting that granted the highest threshold ϵ^*, computed with the tool of protograph DDE. However, to avoid taking false steps, we have softened this approach, by choosing the check splitting that grants the largest level of irregularity while having a sufficiently good threshold. This allows, in the next steps, to have several possibilities for the further check splittings.

In Fig. 2, we plot the inefficiencies f_n computed by (9), using the rate R and the convergence threshold ϵ^* for all members of the designed families, each corresponding to one dot. The leftmost dot of each curve represents the mother code of that family, as listed in Table 1.

Notice that, while check splitting, as a mechanism of rate lowering, does not completely fit in the description of

[3] If the output LLR is 0, we suppose that the VN estimate is random.

[4] Notice that $\mathbf{t}'$ has zero entries in correspondence with the nonzero entries of $\mathbf{t}''$, and vice versa.

Section III, the resulting code $\mathcal{C}_n$ can also be generated by protograph matrix $\mathbf{T}'_n = [\mathbf{T}_{n-1}^{\mathsf{T}}, (\mathbf{t}')^{\mathsf{T}}]^{\mathsf{T}}$. However, protograph $\mathbf{T}_n$ has typically a better BP performance with respect to $\mathbf{T}'_n$, as it has a smaller number of edges. Thus, $\mathbf{T}_n$ has been used to generate the parity-check matrix $\mathbf{H}_n$ for BP decoding.

E. IR BY RATELESS LDPC DECODING

For a given estimated QBER, our proposed schemes identify the most suitable family. The rationale is that, for a given estimate of ϵ, the family with the smallest inefficiency f should be used. Beyond determining the family, the protocol also chooses the best family member to start with, and consequently, the maximum number of family members to use. Once the family and its first member is selected, the corresponding set of syndromes is requested from Bob to Alice. Should decoding fail, Bob requests an additional set of Δ syndromes and tries again to decode. This process goes on until decoding is successful. Note that, even if the real QBER differs from the expected one, reconciliation is not impaired: The key will be reconciled with a slightly larger inefficiency or with a slightly larger number of communication rounds in case of overestimation or underestimation of the real QBER, respectively.

In the optimization of the rateless LDPC scheme, we have considered $\Delta = \alpha S$, $\alpha \leq 1$. For $\alpha = 1$, each further syndrome request from Bob corresponds to switching from a family member to the next one. For $\alpha < 1$, check splitting is performed directly on the rows of the parity-check matrix. From (8), we see that the difference between the rates of two neighboring members of a given family is

$$\Delta R = R_{n-1} - R_n = \frac{\Delta}{N} = \frac{\alpha S}{N_v - N_p} = \frac{\alpha}{\nu_v - \nu_p}. \tag{17}$$

From Table 1, it can be seen that the lower the rate of the mother code, the higher ΔR. Thus, for low-rate families, choosing $\alpha = 1$ would result in a large rate difference between two successive family members. This, in turn, would result in a loss of efficiency whenever, for a given QBER, the starting family member is either too weak (rate too high) or too cautious (rate too low). In order to avoid such shortcomings, choosing $\alpha < 1$ allows a finer granularity of the rate choice by introducing intermediate codes between family members. In that case since at each communication round, a smaller number of syndrome bits is revealed, the scheme will have a lower inefficiency on average but may require a higher number of communication rounds, with respect to the case $\alpha = 1$.

V. NUMERICAL RESULTS

We have implemented our rateless LDPC IR algorithm based on BP decoding with the code families described in Section III-A. Messages have been quantized with 10 b without experiencing any observable penalty with respect to unquantized BP decoding. To give a hint, on a regular PC (Intel i9 7940X, 3.1 GHz CPU), our quantized BP decoder requires on average a computing time per communication round that ranges from 7 ms ($N = 1800$) to approximately 250–300 ms ($N = 90\,000$), roughly proportional to N.

For each QBER in the range [0.7%, 11%],[5] we have simulated the described algorithm over a large number of trials, registering the number of exchanged syndrome bits and of communication rounds needed to reconcile the key. Based on these data, we have computed the average code rate $\overline{R}$ and the average inefficiency $\overline{f}$ according to (10), and the average number of communication rounds. We have repeated the simulation for several possible sifted-key block lengths. The QBER is assumed to be perfectly known. Regarding the value of $\Delta = \alpha S$, we have set $\alpha = 1$ for all families, except for Hp72 and Hp82. For the Hp72 family, we have set $\alpha = 1/2$ while for Hp82, $\alpha = 1/4$. The choice $\alpha < 1$ for these two low-rate families follows from the considerations at the end of Section IV-E.

In Fig. 3, for similar sifted-key block lengths, we show the comparison between the rateless LDPC scheme and their counterparts in [7] and [8]. For the symmetric scheme in [8], the parameter α has the same meaning as for ours. Notice that while we fix the sifted-key block length N in [7] and [8], the schemes are labeled according to the code length N_v. Since $N = N_v - N_p$, from Table 1, the ratio N/N_v will range from 30/31 to 6/7 depending on the family. As an example, $N = 3600$ corresponds to $N_v \in [3720, 4200]$, which is compared to the symmetric scheme with $N_v = 4000$ in Fig. 3. As another example, for $N = 9000$, $N_v \in [9300, 10500]$, compared in the same figure to the blind scheme with $N_v = 10000$. Thus, the difference between the compared values is relatively small.

As can be seen, for $N = 1800$, rateless LDPC codes achieve a much better performance than the blind scheme with $N_v = 2000$. Our proposal also shows a certain improvement with respect to the symmetric scheme with $N_v = 1944$. The rateless scheme with $N = 3600$ performs similarly as the symmetric scheme with $N_v = 4000$, although the latter has a rather bumpy performance curve. Finally, for $N = 9000$, our proposal largely outperforms the blind scheme in [7] for $\epsilon < 2\%$ while the latter shows some advantage for $2\% \leq \epsilon \leq 2.4\%$.

In Fig. 4, we show the comparison in terms of average number of communication rounds. We do not report curves for the blind scheme in [7] since the corresponding curves shown in Fig. 3 are obtained with 200 communication rounds. As Fig. 4 shows, rateless LDPC codes allow us to obtain comparable or even better performance while requiring at most 4 communication rounds on the average, for $N = 1800$, and about 3 for $N = 3600$. Comparing with the symmetric scheme in [8] with similar length, it means a saving from 33% to 67% of the communication rounds, depending on the QBER and the sifted-key block length.

In Fig. 5, we plot the average efficiency $\overline{f}$ versus the QBER ϵ for each sifted-key block length N considered. Notice that,

[5] There is no interest in testing with QBER>11% which is approximately the maximum QBER at which BB84 can generate a secure key.

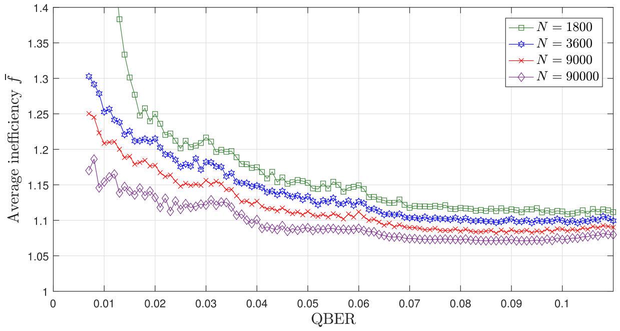

FIGURE 5. Average inefficiency $\bar{f}$ with the proposed rateless LDPC scheme, as a function of QBER, for several values of the sifted-key length N.

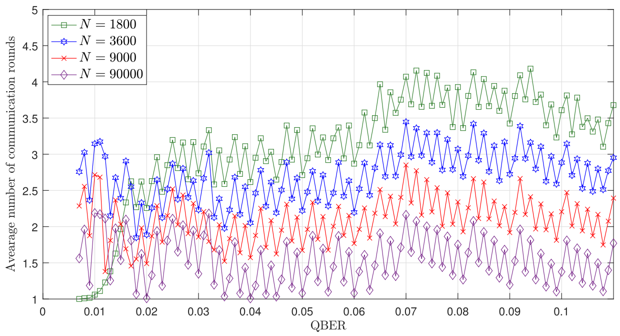

FIGURE 6. Average number of communication rounds needed to reconcile the key with the proposed rateless LDPC scheme, as a function of QBER for several values of the sifted-key length N.

the larger N, the lower the inefficiency, which gets closer and closer to the threshold inefficiency plotted in Fig. 2. For the shortest sifted keys, the scheme becomes less and less efficient at low QBER, due to the fact that the range of available families is smaller, as indicated in Table 2.

In Fig. 6, we plot the average number of communication rounds versus the QBER ϵ. The key length and, in all cases, $\overline{n}$ is very low. Increasing N generally implies a lower variance of the number of communication rounds, due to the fact that a large block length corresponds to a, so to speak, more deterministic behavior of the decoder, as statistical fluctuations tend to fade out. If the starting family member is properly chosen, the smaller variance implies also a smaller average number of communication rounds, as most of the times, one or two communication rounds will be enough to decode. Regarding the behavior for $N = 1800$, it will then be no surprise that it has almost always the largest average number of communication rounds. Since the highest-rate families (Hp32, Hp22, Hp92) are not designed for $N = 1800$, for low QBER, we are forced to use rates lower than those strictly needed, thus raising the average inefficiency, as shown in Fig. 5. In return, those too cautious codes are able to decode already at the first step, thus explaining the minimum average number of communication rounds for $N = 1800$ at low QBER. Finally, the sawtooth appearance in these plots is due to the switch to a different family member depending on the QBER value.

It is worth highlighting that, thanks to the rateless strategy, no residual word error rate has been observed in any of the simulated cases, with a number of reconciled keys ranging from 10^4 to 10^6 depending on the QBER and on the sifted key length N.

VI. CONCLUSION

In this article, we have described a new scheme for IR for QKD applications, based on rateless protograph LDPC codes. This new method has the advantage of flexibility in terms of sifted-key length, which is directly inherited from protograph LDPC codes. The design of the scheme has taken advantage from protograph DDE, which is a novel contribution of this article. Rateless protograph LDPC codes compare favorably with other schemes from the literature, both in terms of average inefficiency and in terms of average number of communication rounds. Thus, for the whole range of useful QBERs, rateless protograph LDPC codes can represent a key factor for the improvement of current QKD systems.

APPENDIX A
BP DECODING

Given an LDPC with parity-check matrix $\mathbf{H}$, consider the transmission of a binary sequence $\mathbf{x}$ with syndrome $\mathbf{s} = \mathbf{Hx}$ on a BSC with error probability ϵ. BP decoding of the received sequence $\mathbf{y}$ can be implemented as follows.

Define the following.

1) $L_{j,\text{ch}}$ as the channel log-likelihood ratio (LLR) for the unpunctured VN v_j, given by

$$L_{j,\text{ch}} = \log \frac{\mathbb{P}\{y_j|x_j = 0\}}{\mathbb{P}\{y_j|x_j = 1\}} = (-1)^{y_j} \log \frac{1-\epsilon}{\epsilon} \qquad (18)$$

and $L_{j,\text{ch}} = 0$ for the punctured VNs, which are not transmitted and for which there is no information received from the channel.

2) $\mathcal{N}_c(i)$ (respectively, $\mathcal{N}_v(j)$) as the set of VNs neighbors of c_i (respectively, the set of CNs neighbors of v_j).

3) $L_{i\leftarrow j}(\ell)$ (respectively, $L_{i\rightarrow j}(\ell)$) as the message from v_j to c_i (respectively, from c_i to v_j) at iteration ℓ.

At the beginning, all CN messages are initialized to zero, i.e., $L_{i\rightarrow j}(0) = 0$, for all i and j. At the ℓth decoding iteration $\ell = 1, 2, \ldots$, VN v_i sends to its neighbor c_j the message

$$L_{i\leftarrow j}(\ell) = L_{j,\text{ch}} + \sum_{c_{i'} \in \mathcal{N}_v(j)\setminus c_i} L_{i'\rightarrow j}(\ell - 1). \qquad (19)$$

Then, c_i replies to v_j with the message

$$L_{i\rightarrow j}(\ell) = (-1)^{s_i} \mathop{\boxtimes}_{v_{j'} \in \mathcal{N}_c(i)\setminus v_j} L_{i\leftarrow j'}(\ell) \qquad (20)$$

where we have

$$L_1 \otimes L_2 = 2\tanh^{-1}\left(\tanh\frac{L_1}{2}\tanh\frac{L_2}{2}\right). \qquad (21)$$

The output LLR for v_j at iteration ℓ is given by

$$L_j(\ell) = L_{j,\mathrm{ch}} + \sum_{c_i \in \mathcal{N}_v(j)} L_{i\to j}(\ell). \tag{22}$$

If the sign of $L_j(\ell)$ is positive, then the hard estimate of x_j, namely $\hat{x}_j$, at iteration ℓ is 0, otherwise is 1. If at a given iteration hard estimates form a codeword (a fact that is easily checked by verifying that $\mathbf{H}\hat{\mathbf{x}} = \mathbf{s}$), the BP decoder has converged, and iterations stop. Otherwise, a new iteration is run. When a predefined maximum number of iterations is reached without convergence, a decoding failure is declared.

APPENDIX B
PROTOGRAPH BP DECODING

In this section, we generalize, to the protograph case, the BP decoding algorithm in quantized form. Let $\mathcal{Q}(\cdot)$ denote the quantization function on $2M+1$ levels $q_{-M}, \ldots, q_M$ symmetric with respect to the origin, so that $q_0 = 0$ and $q_{-m} = -q_m$. We use the same notation of Section III-A.

Given the channel output y_j, the channel LLR for VN ω_j will be given by

$$L_{j,\mathrm{ch}} = \begin{cases} \mathcal{Q}\left((-1)^{y_j} \log \frac{1-\epsilon}{\epsilon}\right), & \text{if } \omega_j \text{ is unpunctured} \\ 0, & \text{if } \omega_j \text{ is punctured.} \end{cases} \tag{23}$$

At the beginning, all CN messages are initialized to zero, i.e., $L_{i\to j}(0) = 0$, for all i and j. Let $t_{i,j}$ represent the number of parallel edges connecting VN ω_i to CN κ_j. At the ℓth decoding iteration, $\ell = 1, 2, \ldots$, VN ω_i sends to κ_j the message

$$\begin{aligned} L_{i\leftarrow j}(\ell) = \mathcal{Q}\Bigg(& L_{j,\mathrm{ch}} + (t_{ij}-1)L_{i\to j}(\ell-1) \\ & + \sum_{\kappa_{i'} \in \mathcal{N}_v(j)\setminus\kappa_i} t_{i'j} L_{i'\to j}(\ell-1)\Bigg). \end{aligned} \tag{24}$$

Then, κ_i replies to ω_j with the message

$$L_{i\to j}(\ell) = L_{i\leftarrow j}(\ell)^{\otimes_q (t_{ij}-1)} \otimes_q \mathop{\prod\hspace{-0.85em}\otimes_q}_{v_{j'} \in \mathcal{N}_c(i)\setminus v_j} L_{i\leftarrow j'}(\ell)^{\otimes_q t_{ij'}} \tag{25}$$

where

$$L_1 \otimes_q L_2 = \mathcal{Q}\left(2\tanh^{-1}\left(\tanh\frac{L_1}{2}\tanh\frac{L_2}{2}\right)\right). \tag{26}$$

The output LLR for v_j at iteration ℓ is given by

$$L_j(\ell) = \mathcal{Q}\left(L_{j,\mathrm{ch}} + \sum_{\kappa_i \in \mathcal{N}_v(j)} t_{ij} L_{i\to j}(\ell)\right). \tag{27}$$

APPENDIX C
UPDATE OF PMFS FOR CN AND VN MESSAGES IN PROTOGRAPH DDE

In this appendix, we give the details of the DDE, as described in Section IV-B. The analysis is based on the computation of the pmf of the messages output by each node, given the pmfs of the input messages and the node type (VN or CN).

Let us first consider the VN-to-CN pmf update, shown in (12). As the VN simply performs the sum of the channel LLR and the incoming messages, the output pmf of function $\mathcal{V}_{ij}$ will be the convolution of the input pmfs, followed by a clipping operation $\mathcal{S}$, as the output pmf must have the same size-$(2M+1)$ support of the input pmfs. In formulas

$$\begin{aligned} \mathcal{P}_{i\leftarrow j}(\ell) &= \mathcal{V}_{ij}\left(\mathcal{P}_{j,\mathrm{ch}}, \{\mathcal{P}_{i'\to j}(\ell-1)\}_{\kappa_{i'} \in \mathcal{N}_v(j)}\right) \\ &= \mathcal{S}\Bigg(\mathcal{P}_{j,\mathrm{ch}} * \mathcal{P}_{i\to j}(\ell-1)^{*(t_{ij}-1)} * \\ &\qquad \mathop{\prod\hspace{-0.85em}\ast}_{\kappa_{i'} \in \mathcal{N}_v(j)\setminus\kappa_i} \mathcal{P}_{i'\to j}(\ell-1)^{*t_{i'j}}\Bigg) \end{aligned} \tag{28}$$

where we have defined the discrete convolution

$$(\mathcal{P}_1 * \mathcal{P}_2)[m] = \sum_{n=\max\{-M_1, m+M_2\}}^{\min\{M_1, m+M_2\}} \mathcal{P}_1[n]\mathcal{P}_2[m-n] \tag{29}$$

for $m = -M_1 - M_2, \ldots, M_1 + M_2$, if $\mathcal{P}_i$ has support $[-M_i, M_i]$, $i = 1, 2$. The saturation function $\mathcal{S}$ truncates the tails of its argument pmf. More precisely, let $\mathcal{P}$ be a pmf defined on discrete support $\{-M', \ldots, M'\}$, with $M' > M$, and let $\mathcal{P}' = \mathcal{S}(\mathcal{P})$. Then

$$\mathcal{P}'[m] = \begin{cases} \sum_{m'=-M'}^{-M} \mathcal{P}[m'], & m = -M \\ \mathcal{P}[m], & -M+1 \le m \le M-1 \\ \sum_{m'=M}^{M'} \mathcal{P}[m'], & m = M \end{cases}. \tag{30}$$

Analogously, the pmf of the output LLR given in (14) can be written as

$$\begin{aligned} \mathcal{P}_j(\ell) &= \mathcal{O}_j\left(\mathcal{P}_{j,\mathrm{ch}}, \{\mathcal{P}_{i\to j}(\ell)\}_{\kappa_i \in \mathcal{N}_v(j)}\right) \\ &= \mathcal{S}\left(\mathcal{P}_{j,\mathrm{ch}} * \mathop{\prod\hspace{-0.85em}\ast}_{\kappa_i \in \mathcal{N}_v(j)} \mathcal{P}_{i\to j}(\ell-1)^{*t_{ij}}\right). \end{aligned} \tag{31}$$

We now consider the CN-to-VN pmf update, given by (13). Consider input message L_i, with pmf $\mathcal{P}_i$, $i = 1, 2$. Let $L = L_1 \otimes_q L_2$. Its pmf $\mathcal{P}$ satisfies

$$\mathcal{P}[m] = \sum_{\substack{m_1, m_2: \\ m_1 \otimes_q m_2 = m}} \mathcal{P}_1[m_1]\mathcal{P}_2[m_2]. \tag{32}$$

To ease notation, we will summarize the previous relationship with the shorthand notation $\mathcal{P} = \mathcal{P}_1 \circledast \mathcal{P}_2$. Then, we will have

$$\begin{aligned} \mathcal{P}_{i\to j}(\ell) &= \mathcal{C}_{ij}\left(\{\mathcal{P}_{i\leftarrow j'}(\ell)\}_{\omega_{j'} \in \mathcal{N}_c(i)}\right) \\ &= \mathcal{P}_{i\leftarrow j}(\ell)^{\circledast(t_{ij}-1)} \circledast \mathop{\prod\hspace{-0.85em}\circledast}_{v_{j'} \in \mathcal{N}_c(i)\setminus v_j} \mathcal{P}_{i\leftarrow j'}(\ell)^{\circledast t_{ij'}}. \end{aligned} \tag{33}$$

REFERENCES

[1] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput., Syst., Signal Process.*, 1984, pp. 175–179, doi: 10.1016/j.tcs.2014.05.025.

[2] G. Brassard and L. Salvail, “Secret-key reconciliation by public discussion,” in *Advances in Cryptology - EUROCRYPT’93*, T. Helleseth, Ed. Berlin, Germany: Springer, 1994, pp. 410–423, doi: 10.1007/3-540-48285-7_35.

[3] J. Martinez-Mateo, C. Pacher, M. Peev, A. Ciurana, and V. Martin, “Demystifying the information reconciliation protocol cascade,” 2014, *arXiv:1407.3257*, doi: 10.48550/arXiv.1407.3257.

[4] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, “Fast, efficient error reconciliation for quantum cryptography,” *Phys. Rev. A*, vol. 67, no. 5, May 2003, Art. no. 052303, doi: 10.1103/PhysRevA.67.052303.

[5] D. Elkouss, A. Leverrier, R. Alleaume, and J. J. Boutros, “Efficient reconciliation protocol for discrete-variable quantum key distribution,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 1879–1883, doi: 10.1109/ISIT.2009.5205475.

[6] E. O. Kiktenko, A. O. Malyshev, and A. K. Fedorov, “Blind information reconciliation with polar codes for quantum key distribution,” *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 79–83, Jan. 2021, doi: 10.1109/LCOMM.2020.3021142.

[7] J. Martinez-Mateo, D. Elkouss, and V. Martin, “Blind reconciliation,” 2013, doi: 10.48550/arXiv.1205.5729.

[8] E. O. Kiktenko, A. S. Trushechkin, C. C. W. Lim, Y. V. Kurochkin, and A. K. Fedorov, “Symmetric blind information reconciliation for quantum key distribution,” *Phys. Rev. Appl.*, vol. 8, 2017, Art. no. 044017, doi: 10.1103/PhysRevApplied.8.044017.

[9] N. Borisov, I. Petrov, and A. Tayduganov, “Asymmetric adaptive LDPC-based information reconciliation for industrial quantum key distribution,” *Entropy*, vol. 25, no. 1, 2023, Art. no. 31, doi: 10.3390/e25010031.

[10] G. Limei, R. Qi, J. Di, and H. Duan, “QKD iterative information reconciliation based on LDPC codes,” *Int. J. Theor. Phys.*, vol. 59, no. 6, 2020, pp. 1717–1729, doi: 10.1007/s10773-020-04438-9.

[11] Z. Liu, Z. Wu, and A. Huang, “Blind information reconciliation with variable step sizes for quantum key distribution,” *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 170, doi: 10.1038/s41598-019-56637-y.

[12] H.-K. Mao, Y.-C. Qiao, and Q. Li, “High-efficient syndrome-based LDPC reconciliation for quantum key distribution,” *Entropy*, vol. 23, no. 11, 2021, Art. no. 1440, doi: 10.3390/e23111440.

[13] S. K. Liao et al., “Satellite-to-ground quantum key distribution,” *Nature*, vol. 549, pp. 43–47, 2017, doi: 10.1038/nature23655.

[14] H.-K. Lo, X. Ma, and K. Chen, “Decoy state quantum key distribution,” *Phys. Rev. Lett.*, vol. 94, Jun. 2005, Art. no. 230504, doi: 10.1103/PhysRevLett.94.230504.

[15] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2008, doi: 10.1017/CBO9780511791338.

[16] S.-Y. Chung, G. Forney, T. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001, doi: 10.1109/4234.905935.

[17] T. V. Nguyen, A. Nosratinia, and D. Divsalar, “The design of rate-compatible protograph LDPC codes,” *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, Oct. 2012, doi: 10.1109/TCOMM.2012.081012.110010.

[18] A. K. Ekert, “Quantum cryptography based on Bell’s theorem,” *Phys. Rev. Lett.*, vol. 67, no. 6, pp. 661–663, 1991, doi: 10.1103/PhysRevLett.67.661.

Alberto Tarable received the Laurea degree (*summa cum laude*) in electronic engineering and the Ph.D. degree in electronic and telecommunication engineering from the Politecnico di Torino, Turin, Italy, in 1998 and 2002, respectively.

From 2002 to 2012, he was a Postdoctoral Researcher with the Department of Electronics and Telecommunications, Politecnico di Torino. Since 2012, he has been a Researcher with the Institute of Electronics, Computer and Telecommunication Engineering, Italian National Research Council (CNR), Rome, Italy. He has coauthored some 70 papers on the subjects of telecommunications, information theory, and computer science. His current research interests include physical-layer design of communication systems, quantum key distribution systems, and crowdsourcing systems.

Rudi Paolo Paganelli received the Dr. Eng. degree in electrical engineering and the Ph.D. degree in electrical engineering, computer science, and telecommunications from the University of Bologna, Bologna, Italy, in 1998 and 2002, respectively.

Since 2002, he has been a Research Fellow with the Institute of Electronics and Information Engineering, National Research Council, Bologna. He is also an Adjunct Professor in analog and power electronics with the University of Bologna. He is a coauthor in several tens of papers in the field of power and microwave circuits CAD modeling and design. His current research interests include computer-assisted circuit designs, power converters, microwave amplifiers, and quantum technologies.

Marco Ferrari (Member, IEEE) was born in Milano, Italy, in 1971. He received the Laurea degree (M.S. equivalent) in telecommunications engineering (*cum laude*) and the Ph.D. degree in electronics and communication engineering from the Politecnico di Milano, Milano, in 1996 and 2000, respectively.

Since 2001, he has been a Researcher with the Istituto di Elettronica e di Ingegneria dell’Informazione e delle Teelcomunicazioni (IEIIT), Consiglio Nazionale delle Ricerche (CNR), Politecnico di Milano. In 2002, he was an EPRSC Research Fellow with the University of Plymouth, U.K. He has coauthored approximately 70 scientific publications in leading international journals and conference proceedings and a few patents. His main research interests include channel coding, information theory, and digital transmission.

Open Access funding provided by ‘Consiglio Nazionale delle Ricerche-CARI-CARE-ITALY’ within the CRUI CARE Agreement